BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-222384
(43)Date of publication of application : 11.08.2000

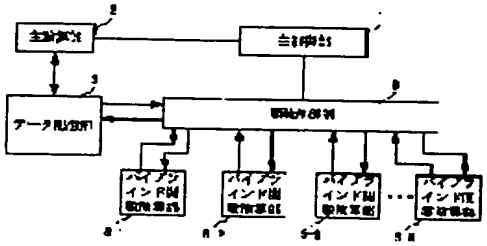
(51)Int.Cl. G06F 17/10
G06F 7/00
G06T 1/20

(21)Application number : 11-022934
(22)Date of filing : 29.01.1999

(71)Applicant : YAMATAKE CORP
(72)Inventor : MORIKAWA MAKOTO

(54) ARITHMETIC PROCESSOR

(57)Abstract:
PROBLEM TO BE SOLVED: To provide an arithmetic processor quickly operating the maximum data processing using plural functions.
SOLUTION: This arithmetic processor is provided with pipe lined function arithmetic parts 5-1-5-N. A function controlling part 6 is provided between a main control part 1 and the pipe lined function arithmetic parts 5-1-5-N, and the pipe lined function arithmetic parts pine lined function arithmetic group) to be executed and the execution sequence are designated from the main control part 1 to the function controlling part 6. The function controlling part 6 reads data to be processed from a data storing part, and inputs the data to the pipe lined function arithmetic group (the pipe lined function arithmetic parts cascade- connected in the designated execution sequence), and the data of the arithmetic processed results from the pipe lined function arithmetic group are written in the data storing part 3.



LEGAL STATUS

[Date of request for examination] 28.09.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2000－222384
（P2000－222384A）
(43)公開日　平成12年８月11日(2000.8.11)

(51)Int.Cl'	識別記号	FI		テーマコード'(参考)
Ｇ０６Ｆ　17/10		Ｇ０６Ｆ　15/31	Ｚ	５Ｂ０２２
7/00	-	7/00	Ａ	５Ｂ０５６
Ｇ０６Ｔ　1/20		15/66	Ｌ	５Ｂ０５７

審査請求　未請求　請求項の数３　ＯＬ　（全 13 頁）

(21)出願番号　特願平11－22934

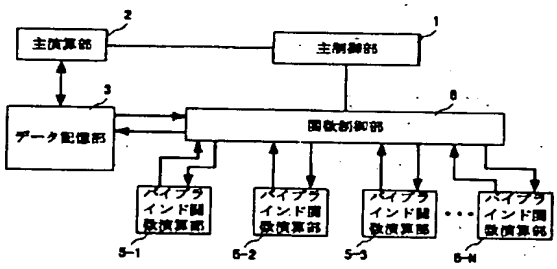
(22)出願日　平成11年１月29日(1999.1.29)

(71)出願人　000006666
株式会社山武
東京都渋谷区渋谷２丁目12番19号
(72)発明者　森川　賦
東京都渋谷区渋谷２丁目12番19号　株式会社山武内
(74)代理人　100064621
弁理士　山川　政樹

Ｆターム(参考)　5B022 AA01 BA00 DA02 FA01
5B056 AA04 BB13 BB21 FF16 HH03
5B057 CG05 CH04 CH05 CH11

(54)【発明の名称】　演算処理装置

(57)【要約】
【課題】　多数の関数を用いる多大なデータ処理を高速で行う。
【解決手段】　パイプラインド関数演算部５－１～５－Ｎを設ける。主制御部１とパイプラインド関数演算部５－１～５－Ｎとの間に関数制御部６を設け、主制御部１より関数制御部６に対して実行すべきパイプラインド関数演算部（パイプラインド関数演算群）およびその実行順序を指定する。関数制御部６は、データ記憶部から処理すべきデータを読み出し、パイプラインド関数演算群（指定の実行順序で縦続接続されたパイプラインド関数演算部）に投入し、このパイプラインド関数演算群からの演算処理結果のデータをデータ記憶部３に書き込む。



【特許請求の範囲】
【請求項１】　処理すべきデータが格納されたデータ記憶部と、
入力データの演算処理を終了する前に新たな入力データの演算処理を開始することの可能な複数のパイプライン関数演算部と、
これらパイプライン関数演算部の中から実行すべきパイプライン関数演算部をパイプライン関数演算群として選択のうえその実行順序を指定する主制御部と、
この主制御部によって指定された実行順序に従って前記パイプライン関数演算群のパイプライン関数演算部を縦続接続させ、その処理データの受け渡しを制御する一方、前記データ記憶部から処理すべきデータを読み出して前記パイプライン関数演算群に入力すると共にこのパイプライン関数演算群からの演算処理結果のデータを前記データ記憶部に書き込む関数制御部とを備えたことを特徴とする演算処理装置。
【請求項２】　処理すべきデータが格納された第１のデータ記憶部と、
処理すべきデータが格納される第２のデータ記憶部と、
入力データの演算処理を終了する前に新たな入力データの演算処理を開始することの可能な複数のパイプライン関数演算部と、
前記第２のデータ記憶部から処理すべきデータを読み出し、この読み出したデータに対して前記パイプライン関数演算部で行う演算とは別個の主演算を行い、その演算処理結果のデータを前記第２のデータ記憶部に書き込む主演算部と、
前記パイプライン関数演算部の中から前記第１のデータ記憶部に格納されている処理すべきデータに対して実行すべきパイプライン関数演算部を第１のパイプライン関数演算群として選択のうえその実行順序を指定する一方、前記第２のデータ記憶部に格納されている前記主演算部での演算処理結果のデータに対して実行すべきパイプライン関数演算部を第２のパイプライン関数演算群として選択のうえその実行順序を指定する主制御部と、
この主制御部によって指定された実行順序に従って前記第１のパイプライン関数演算群のパイプライン関数演算部を縦続接続させ、その処理データの受け渡しを制御する一方、前記第１のデータ記憶部から処理すべきデータを読み出して前記第１のパイプライン関数演算群に入力すると共にこの第１のパイプライン関数演算群からの演算処理結果のデータを前記主演算部で処理すべきデータとして前記第２のデータ記憶部に書き込む機能と、前記主制御部によって指定された実行順序に従って前記第２のパイプライン関数演算群のパイプライン関数演算部を縦続接続させ、その処理データの受け渡しを制御する一方、前記演算部での演算処理結果のデータを前記第２のデータ記憶部から読み出して前記第２のパイプライン関数演算群に入力すると共にこの第２のパイプライン関数演算群からの演算処理結果のデータを前記第１のデータ記憶部に書き込む機能とを有する関数制御部とを備えたことを特徴とする演算処理装置。
【請求項３】　処理すべきデータが格納された第１のデータ記憶部と、
処理すべきデータが格納される第２および第３のデータ記憶部と、
入力データの演算処理を終了する前に新たな入力データの演算処理を開始することの可能な複数のパイプライン関数演算部と、
前記第２および第３のデータ記憶部から処理すべきデータをデータ順に応じて互い違いに読み出し、この読み出したデータに対して前記パイプライン関数演算部で行う演算とは別個の主演算を行い、その演算処理結果のデータを読み出し元のデータ記憶部に書き込む主演算部と、
前記パイプライン関数演算部の中から前記第１のデータ記憶部に格納されている処理すべきデータに対して実行すべきパイプライン関数演算部を第１のパイプライン関数演算群として選択のうえその実行順序を指定する一方、前記第２および第３のデータ記憶部に格納されている前記主演算部での演算処理結果のデータに対して実行すべきパイプライン関数演算部を第２のパイプライン関数演算群として選択のうえその実行順序を指定する主制御部と、
この主制御部によって指定された実行順序に従って前記第１のパイプライン関数演算群のパイプライン関数演算部を縦続接続させ、その処理データの受け渡しを制御する一方、前記第１のデータ記憶部から処理すべきデータを読み出して前記第１のパイプライン関数演算群に入力すると共にこの第１のパイプライン関数演算群からの演算処理結果のデータを前記主演算部で処理すべきデータとして前記第２および第３のデータ記憶部にデータ順に応じて互い違いに書き込む機能と、前記主制御部によって指定された実行順序に従って前記第２のパイプライン関数演算群のパイプライン関数演算部を縦続接続させ、その処理データの受け渡しを制御する一方、前記主演算部での演算処理結果のデータを前記第第２および第３のデータ記憶部からデータ順に応じて互い違いに読み出して前記第２のパイプライン関数演算群に入力すると共にこの第２のパイプライン関数演算群からの演算処理結果のデータを前記第１のデータ記憶部に書き込む機能とを有する関数制御部とを備えたことを特徴とする演算処理装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】この発明は、画像処理などに用いて好適な演算処理装置に関するものである。
【０００２】

【従来の技術】本出願人が開発を進めている画像処理装置では、画像データに対して拡大，縮小，回転，ＦＦＴ（高速フーリエ変換），ＤＣＴ（離散コサイン変換），相関計算などの主演算の他に、オフセット除去などの前処理や正規化処理などの後処理を行っている。例えば、位相限定方式のパターンマッチング装置等で、ＦＦＴ処理を行う前に画像データにオフセット分を除去する前処理や、複素数を位相情報に変換する後処理を行っている。

【０００３】図６にこの画像処理装置の要部構成を示す。同図において、１は主制御部、２はＦＦＴ処理を行う主演算部、３はデータ記憶部、４－１～４－Ｎは関数演算部である。データ記憶部３には処理すべき多数のデータが格納されている。関数演算部４－１～４－Ｎは現在演算中の入力データの演算処理を終了してからでないと新たな入力データの演算処理を開始することはできない。

【０００４】この画像処理装置では次のようにしてＦＦＴ処理が行われる。主制御部１は、主演算部２でのＦＦＴ処理の前に、データ記憶部３に格納されている処理すべきデータに対して関数演算部４－１～４－Ｎを利用して前処理を行わせる。この場合、主制御部１は、関数演算部４－１～４－Ｎの中から実行すべき関数演算部を制御信号線Ｓ１～ＳＮを介して選択する。ここでは、例えば、その実行順序を４－１→４－２→４－３として、関数演算部４－１，４－２，４－３を選択するものとする。

【０００５】次に、主制御部１は、データ記憶部３から１番目のデータを読み出し、データバスＤＢを介して関数演算部４－１へ送る。関数演算部４－１での演算処理が終了すれば、その演算処理結果をデータバスＤＢを介してデータ記憶部３へ書き込んだうえ、データバスＤＢを介して関数演算部４－２へ送る。関数演算部４－２での演算処理が終了すれば、その演算処理結果をデータバスＤＢを介してデータ記憶部３へ書き込んだうえ、データバスＤＢを介して関数演算部４－３へ送る。関数演算部４－３での演算処理が終了すれば、その演算処理結果のデータを前処理完了データとしてデータ記憶部３に書き込む。そして、２番目のデータをデータ記憶部３から読み出し、１番目のデータと同様の演算シーケンスを施し、処理すべき全てのデータが完了するまでこの演算シーケンスを繰り返す。

【０００６】なお、１つのデータの前処理が完了（データ記憶部３への書き込み）してから、次のデータの前処理を開始（関数演算部４－１へのデータの供与）する理由の１つとして、複数の関数演算部４が同時にアクティブになると、データバスＤＢ上でデータ同士の衝突が発生し動作できなくなる虞れがあることが挙げられる。

【０００７】Ｋ個のデータの前処理が完了すると、主演算部２は、主制御部１の指示に従い、データ記憶部３に格納されている前処理完了データを読み出し、この前処理完了データに対してＦＦＴ処理を施し、このＦＦＴ処理を施したデータ（ＦＦＴ完了データ）をデータ記憶部３に書き込む。全てのデータのＦＦＴ処理が完了すると、主制御部１は、データ記憶部３に格納されているＦＦＴ完了データに対し、関数演算部４－１～４－Ｎの中から実行すべき関数演算部およびその実行順序を指定のうえ、前処理と同様にして後処理を行わせる。

【０００８】

【発明が解決しようとする課題】しかしながら、このような従来の画像処理装置では、前処理や後処理に必要な関数演算部が増える毎に、（個々の関数の処理時間）×処理データ数（通常は処理対象の画素数）の処理時間が加算されるため、処理時間が比例して増加してしまい、多数の関数を用いる画像処理を高速化することが困難であった。

【０００９】例えば、Ｋ個のデータについて前処理を行う場合、同期回路での処理を想定すると、次のような処理の流れになる。なお、以下では、データ記憶部３からのリードおよびライト動作は通常同時には処理できず、実際にはそれぞれ１サイクルずつ加算される場合もあるが、説明の簡略化のため、リードおよびライトサイクルの時間は関数演算部での演算処理時間（サイクル数）Ｓｎに含まれているものとする。

【００１０】データ１：リード（０）→関数演算部４－１での演算処理（Ｓ１）→関数演算部４－２での演算処理（Ｓ１＋Ｓ２）→関数演算部４－３での演算処理（Ｓ１＋Ｓ２＋Ｓ３）→データ１：ライト（Ｓ１＋Ｓ２＋Ｓ３）→データ２：リード（Ｓ１＋Ｓ２＋Ｓ３）→関数演算部４－１での演算処理（２Ｓ１＋Ｓ２＋Ｓ３）→関数演算部４－２での演算処理（２Ｓ１＋２Ｓ２＋Ｓ３）→関数演算部４－３での演算処理（２Ｓ１＋２Ｓ２＋２Ｓ３）・・・・データＫ：ライト（Ｋ・（Ｓ１＋Ｓ２＋Ｓ３））

【００１１】すなわち、この例では、Ｋ個のデータについて前処理を行う場合、Ｋ・（Ｓ１＋Ｓ２＋Ｓ３）の処理時間を必要とする。ここで、ｎ個の関数を実行する必要があり、それぞれの関数での演算処理時間をＳｎ、処理する必要のあるデータ数をＫとすると、従来タイプの画像処理装置での前処理時間（あるいは後処理時間）Ｔ０は、下記（１）式で表される。

【００１２】

【数１】

$$T0 = K \cdot \sum_{i=1}^{n} Sn \qquad \cdots\cdots (1)$$

【００１３】このような式になるため、従来においては、前処理や後処理の関数で演算処理時間がかかるものや、処理関数の個数に比例して、全体の処理時間が増加してしまい、画像データのようにデータが多大で様々な演算処理を行う画像処理装置の高速化を実現することが

困難となっていた。

【００１４】本発明はこのような課題を解決するためになされたもので、その目的とするところは、多数の関数を用いる多大なデータ処理を高速で行うことの可能な演算処理装置を提供することにある。

【００１５】

【課題を解決するための手段】このような目的を達成するために、第１発明（請求項１に係る発明）は、データ記憶部と、複数のパイプラインド関数演算部と、主制御部と、関数制御部とを設け、パイプラインド関数演算部の中から実行すべきパイプラインド関数演算部をパイプラインド関数演算群として選択のうえその実行順序を指定するものとし、この指定された実行順序に従ってパイプラインド関数演算群のパイプラインド関数演算部を縦続接続させ、データ記憶部から処理すべきデータを読み出してパイプラインド関数演算群に入力すると共にこのパイプラインド関数演算群からの演算処理結果のデータをデータ記憶部に書き込むようにしたものである。

【００１６】この発明によれば、データ記憶部から処理すべきデータが読み出され、パイプラインド関数演算群に入力され、このパイプラインド関数演算群からの演算処理結果のデータがデータ記憶部に書き込まれる。ここで、パイプラインド関数演算群は、そのパイプラインド関数演算群を構成するパイプラインド関数部およびその実行順序が主制御部により指定され、この指定された実行順序に従ってパイプラインド関数演算群のパイプラインド関数演算部が縦続接続される。この縦続接続されたパイプラインド関数演算部の各々は、現在演算中の入力データの演算処理を終了する前に新たな入力データの演算処理を開始することが可能であり、パイプライン関数演算部での処理時間Ｓｎより遥かに小さい投入待ち時間Ｘ（Ｘ＜＜Ｓｎ）でもってパイプラインド関数演算群へ次々にデータを入力することができる。

【００１７】第２発明（請求項２に係る発明）は、第１のデータ記憶部と、第２のデータ記憶部と、複数のパイプラインド関数演算部と、主演算部と、主制御部と、関数制御部とを設け、パイプラインド関数演算部の中から第１のデータ記憶部に格納されている処理すべきデータに対して実行すべきパイプラインド関数演算部を第１のパイプラインド関数演算群として選択のうえその実行順序を指定するものとし、この指定された実行順序に従って第１のパイプラインド関数演算群のパイプラインド関数演算部を縦続接続させ、第１のデータ記憶部から処理すべきデータを読み出して第１のパイプラインド関数演算群に入力すると共にこの第１のパイプラインド関数演算群からの演算処理結果のデータを主演算部で処理すべきデータとして第２のデータ記憶部に書き込むようにし、この第２のデータ記憶部から処理すべきデータを読み出し、この読み出したデータに対して主演算を行い、その演算処理結果のデータを第２のデータ記憶部に書き込むようにし、また、第２のデータ記憶部に格納されている主演算部での演算処理結果のデータに対して実行すべきパイプラインド関数演算部を第２のパイプラインド関数演算群として選択のうえその実行順序を指定するものとし、この指定された実行順序に従って第２のパイプラインド関数演算群のパイプラインド関数演算部を縦続接続させ、主演算部での演算処理結果のデータを第２のデータ記憶部から読み出して第２のパイプラインド関数演算群に入力すると共にこの第２のパイプラインド関数演算群からの演算処理結果のデータを第１のデータ記憶部に書き込むようにしたものである。

【００１８】この発明によれば、第１のデータ記憶部から処理すべきデータが読み出され、第１のパイプラインド関数演算群に入力され、この第１のパイプラインド関数演算群からの演算処理結果のデータが第２のデータ記憶部に書き込まれる。そして、この第２のデータ記憶部に書き込まれたデータが読み出されて主演算が行われ、その演算処理結果のデータが第２のデータ記憶部に書き込まれる。この第２のデータ記憶部に書き込まれた主演算部での演算処理結果のデータは、第２のパイプラインド関数演算群に入力され、この第２のイプラインド関数演算群からの演算処理結果のデータが第１のデータ記憶部に書き込まれる。

【００１９】ここで、第１および第２のパイプラインド関数演算群は、そのパイプラインド関数演算群を構成するパイプラインド関数部およびその実行順序が主制御部により指定され、この指定された実行順序に従ってパイプラインド関数演算群のパイプラインド関数演算部が縦続接続される。この縦続接続されたパイプラインド関数演算部の各々は、現在演算中の入力データの演算処理を終了する前に新たな入力データの演算処理を開始することが可能であり、パイプライン関数演算部での処理時間Ｓｎより遥かに小さい投入待ち時間Ｘ（Ｘ＜＜Ｓｎ）でもって第１および第２のパイプラインド関数演算群へ次々にデータを入力することができる。また、第２のデータ記憶部では主演算部との間でデータの読み出しと書き込みとが短時間で切り替わる可能性があるが、第１のデータ記憶部ではデータの読み出しと書き込みとが短時間で切り替わらない。

【００２０】第３発明（請求項３に係る発明）は、第１のデータ記憶部と、第２のデータ記憶部と、第３のデータ記憶部と、複数のパイプラインド関数演算部と、主演算部と、主制御部と、関数制御部とを設け、パイプラインド関数演算部の中から第１のデータ記憶部に格納されている処理データに対して実行すべきパイプラインド関数演算部を第１のパイプラインド関数演算群として選択のうえその実行順序を指定するものとし、この指定された実行順序に従って第１のパイプラインド関数演算群のパイプラインド関数演算部を縦続接続させ、第１のデータ記憶部から処理すべきデータを読み出して第１のパイ

プラインド関数演算群に入力すると共にこの第１のパイプラインド関数演算群からの演算処理結果のデータを主演算部で処理すべきデータとして第２および第３のデータ記憶部にデータ順に応じて互い違いに書き込むようにし、この第２および第３のデータ記憶部から処理すべきデータをデータ順に応じて互い違いに読み出し、この読み出したデータに対して主演算を行い、その演算処理結果のデータを読み出し元のデータ記憶部に書き込むようにし、また、第２および第３のデータ記憶部に格納されている主演算部での演算処理結果のデータに対して実行すべきパイプラインド関数演算部を第２のパイプラインド関数演算群として選択のうえその実行順序を指定するものとし、この指定された実行順序に従って第２のパイプラインド関数演算群のパイプラインド関数演算部を縦続接続させ、主演算部での演算処理結果のデータを第２および第３のデータ記憶部からデータ順に応じて互い違いに読み出して第２のパイプラインド関数演算群に入力すると共にこの第２のパイプラインド関数演算群からの演算処理結果のデータを第１のデータ記憶部に書き込むようにしたものである。

【００２１】この発明によれば、第１のデータ記憶部から処理すべきデータが読み出され、第１のパイプラインド関数演算群に入力され、この第１のパイプラインド関数演算群からの演算処理結果のデータがデータ順に応じて互い違いに第２および第３のデータ記憶部に書き込まれる。また、第１および第２のデータ記憶部に書き込まれたデータがデータ順に応じて互い違いに読み出されて主演算部による演算が行われ、この主演算部での演算処理結果のデータが読み出し元のデータ記憶部に書き込まれる。第１および第２のデータ記憶部に書き込まれた主演算部での演算処理結果のデータは、データ順に応じて互い違いに第２のパイプラインド関数演算群に入力され、この第２のパイプラインド関数演算群からの演算処理結果のデータが第１のデータ記憶部に書き込まれる。

【００２２】ここで、第１および第２のパイプラインド関数演算群は、そのパイプラインド関数演算群を構成するパイプラインド関数部およびその実行順序が主制御部により指定され、この指定された実行順序に従って第１および第２のパイプラインド関数演算群のパイプラインド関数演算部が縦続接続される。この縦続接続されたパイプラインド関数演算部の各々は、現在演算中の入力データの演算処理を終了する前に新たな入力データの演算処理を開始することが可能であり、パイプライン関数演算部での処理時間Ｓｎより遥かに小さい投入待ち時間Ｘ（Ｘ＜＜Ｓｎ）でもって第１および第２のパイプラインド関数演算群へ次々にデータを入力することができる。

【００２３】また、第２および第３のデータ記憶部では主演算部との間でデータの読み出しと書き込みとが短時間で切り替わる可能性があるが、第１のデータ記憶部ではデータの読み出しと書き込みとが短時間で切り替わらない。また、主演算部と第３のデータ記憶部（第２のデータ記憶部）との間で主演算処理を行っている間に、第２のデータ記憶部（第３のデータ記憶部）からの第２のパイプラインド演算関数群を介する第１のデータ記憶部への演算処理結果のデータの書き込み、第１のデータ記憶部からの第１のパイプラインド演算関数群を介する第２のデータ記憶部（第３のデータ記憶部）への演算処理結果のデータの書き込みを行うことができる。

【００２４】

【発明の実施の形態】以下、本発明を実施の形態に基づき詳細に説明する。

〔実施の形態１：第１発明〕図１はこの発明の一実施の形態の要部を示すブロック図である。同図において、図６と同一符号は同一或いは同等構成要素を示し、その説明は省略する。

【００２５】この実施の形態１では、従来の関数演算部４－１～４－Ｎに代えて、１入力１出力のパイプラインド関数演算部５－１～５－Ｎを使用している。また、主制御部１とパイプラインド関数演算部５－１～５－Ｎとの間に関数制御部６を設け、主制御部１より関数制御部６に対して実行すべきパイプラインド関数演算部（パイプラインド関数演算群）およびその実行順序を指定するようにしている。

【００２６】パイプラインド関数演算部５（５－１～５－Ｎ）は、現在演算中の入力データの演算処理を終了する前に新たな入力データの演算処理を開始することが可能な関数演算部（例えば、四則演算などの関数処理用）であり、１番目の処理対象データを入力後、その演算処理結果がまだ出力されていなくても（１データの関数演算に必要な時間Ｓｎが経過していなくても）、すぐ次のサイクルに２番目の処理対象データの入力が可能である。また、このパイプラインド関数演算部５には、ある関数で演算が完了後、次の関数への受け渡しが可能なようにハンドシェーク信号を出力するようなインターフェイスを構成しておく。

【００２７】関数制御部６は、基本的にはマルチプレクサで構成されており、主制御部１によって選択されたパイプラインド関数演算群のパイプラインド関数演算部を指定された実行順序に従って縦続接続させ、その処理データの受け渡しを制御する一方、データ記憶部３から処理すべきデータを読み出してそのパイプラインド関数演算群に入力すると共に、このパイプラインド関数演算群からの演算処理結果のデータをデータ記憶部３に書き込む機能を有している。

【００２８】この画像処理装置では次のようにしてＦＦＴ処理が行われる。主制御部１は、主演算部２でのＦＦＴ処理の前に、データ記憶部３に格納されている処理すべデータに対してパイプラインド関数演算部５－１～５－Ｎを利用して前処理を行わせる。

【００２９】この場合、主制御部１は、関数制御部６に

対して、パイプラインド関数演算部５－１～５－Ｎの中から前処理に際して実行すべきパイプラインド関数演算部を第１のパイプラインド関数演算群として選択のうえ、その実行順序を指定する。ここでは、例えば、パイプラインド関数演算部５－１，５－２，５－３を第１のパイプラインド関数演算群として選択し、その実行順序を５－１→５－２→５－３とするものとする。

【００３０】関数制御部６は、この主制御部１によって指定された実行順序に従って第１のパイプラインド関数演算群のパイプラインド関数演算部５－１，５－２，５－３を縦続接続させ、データ記憶部３から処理すべき１番目のデータを読み出して第１のパイプラインド関数演算群に入力する。この第１のパイプラインド関数演算群に入力されたデータは、パイプラインド関数演算部５－１→５－２→５－３の順に演算処理され、最後のパイプラインド関数演算部５－３からの演算処理結果のデータが前処理完了データとしてデータ記憶部３に書き込まれる。

【００３１】ここで、関数制御部６は、データ記憶部３から処理すべき１番目のデータを読み出した後、次のサイクルで２番目のデータを読み出して第１のパイプラインド関数演算群に投入する。すなわち、データ記憶部３から１番目のデータを読み出して第１のパイプラインド関数演算群に投入した後、パイプラインド関数演算部での処理時間Ｓｎより遥かに短い投入待ち時間Ｘ（Ｘ＜＜Ｓｎ）でもって、データ記憶部３から２番目のデータを読み出して第１のパイプラインド関数演算群に投入する。

【００３２】第１のパイプラインド関数演算群に投入された２番目のデータは、１番目のデータと同様にしてパイプラインド関数演算部５－１→５－２→５－３の順に演算処理され、最後のパイプラインド関数演算部５－３からの演算処理結果のデータが処理完了データとしてデータ記憶部３に書き込まれる。以下、同様の演算シーケンスを施し、処理すべき全てのデータが完了するまでこの演算シーケンスを繰り返す。

【００３３】全てのデータの前処理が完了すると、主演算部２は、主制御部１の指示に従い、データ記憶部３に格納されている前処理完了データを読み出し、この前処理完了データに対してＦＦＴ処理を施し、このＦＦＴ処理を施したデータ（ＦＦＴ完了データ）をデータ記憶部３に書き込む。

【００３４】全てのデータのＦＦＴ処理が完了すると、主制御部１は、関数制御部６に対して、パイプラインド関数演算部５－１～５－Ｎの中から後処理に際して実行すべきパイプラインド関数演算部を第２のパイプラインド関数演算群として選択のうえ、その実行順序を指定する。

【００３５】関数制御部６は、この主制御部１によって選択された第２のパイプラインド関数演算群のパイプラインド関数演算部を指定された実行順序に従って縦続接続させ、前処理と同様にして、データ記憶部３に格納されているＦＦＴ完了データを第２のパイプラインド関数演算群へ次々に投入して後処理を行わせ、その後処理完了データをデータ記憶部３に書き込む。

【００３６】この実施の形態１では、指定された実行順序に従ってパイプラインド関数演算群のパイプラインド関数演算部を縦続接続させ、この縦続接続させたパイプラインド関数演算部にデータを通して行くので、すなわち複数の関数演算が連続的にパイプライン処理されるので、バスの衝突等が発生することがない。このため、パイプラインド関数演算群におけるパイプラインド関数演算部の個数をｎ個とし、それぞれのパイプラインド関数演算部での処理時間をＳｎ、処理すべきデータ数をＫ個とすると、全てのデータをパイプラインド関数演算群によって演算処理する時間（前処理時間／後処理時間）Ｔｎは、下記（２）式で表される。

【００３７】

【数２】

$$Tn=(K-1)\cdot X+\sum_{i=1}^{n}Sn \quad \cdots\cdots (2)$$

【００３８】この（２）式と前述した従来タイプの（１）式とを比較して分かるように、この実施の形態１では、関数の必要数や画素数が増加しても、それに殆ど影響されずに、極めて高速に前処理や後処理を行うことが可能となる。

【００３９】〔実例〕例えば、５１２×５１２画素の画像データがあり、これをＦＦＴ処理（主演算：処理時間２０ｍｓ）する場合を考えてみる。ＦＦＴ処理前に、予めオフセット分を除去し（前処理関数１：減算、処理時間４０ｎｓ）、それを増幅するものとする（前処理関数２：乗算、処理時間８０ｎｓ）。また、ＦＦＴ処理結果の最大値を用いて正規化し（後処理関数１：除算、処理時間３８０ｎｓ）、ある値以下を０にする（後処理関数２：しきい値処理、処理時間２０ｎｓ）ものとする。これらの個々の処理関数の処理時間は、従来タイプの場合も本実施の形態の場合（新タイプ）も、同じ時間とする。また、主演算処理も同処理時間とする。

【００４０】ここで、従来タイプも新タイプも、共に動作周波数５０ＭＨｚ（１サイクル＝２０ｎｓ）と仮定すると、それぞれの全処理時間ＴＡおよびＴＢは次のようになる。

【００４１】〔従来タイプ：ＴＡ〕

前処理：５１２×５１２×（４０＋８０）ｎｓ≒３１．５ｍｓ

主演算：２０ｍｓ

後処理：５１２×５１２×（３８０＋２０）ｎｓ≒１０

4．9ms
TA=31．5ms+20ms+104．9ms=156．4ms
【0042】〔新タイプ：TB〕
前処理：（512×512−1）×20ns+（40+80）ns≒5．24ms
主演算：20ms
後処理：（512×512−1）×20ns+（380+20）ns≒5．24ms
TB=5．24ms+20ms+5．24ms=30．5ms
【0043】この場合、従来タイプの全処理時間TAと新タイプの全処理時間TBとの差はTA−TB=125．9msであり、新タイプの方が従来タイプよりも5倍以上高速に処理できることが確認できる。
【0044】ここで、特筆すべきは点は前処理および後処理に要する時間で、新タイプでは従来タイプに比べ、6〜20倍以上高速に処理可能となっている。この実例では、前処理よりも後処理の方が処理時間にして3．3倍大きいため、従来タイプでは後処理の方が3．3倍処理時間を必要としている。一方、新タイプでは、縦続接続可能なインターフェイスを持つパイプラインド関数演算部を用いた処理構成のため、殆ど処理時間の増加にはつながっていない。これは処理対象データ（ここでは、512×512）が多いほど、その影響度は少なくなる。
【0045】関数単体の処理時間が増加しても影響が少ないのと同様に、処理関数の増加に対しても新タイプでは影響が少ない。そのため、新タイプでは、複雑な多数の関数処理が必要となるような前処理や後処理でも、処理時間の増加を殆ど伴わずに実行可能である。
【0046】〔実施の形態2〕実施の形態1では、データ記憶部3において、データの読み出しと書き込みとが交互に頻繁に切り替えられる。このため、データ記憶部3としては、データの読み出しと書き込みとの切り替えに際して殆ど待ち時間の生じない高性能の大容量メモリを使用する必要があり、高価となる。データの読み出しと書き込みとの切り替えに際して比較的待ち時間が生じる安価な大容量メモリをデータ記憶部3として使用すると、処理速度の劣化につながり、高速処理が阻害される。
【0047】そこで、この実施の形態2では、データ記憶部3として高性能の大容量メモリを使用しなくてもよいようにして（データ記憶部3として比較的待ち時間が生じる安価な大容量メモリの使用を可能として）、コストパフォーマンスの向上を図る。
【0048】図2はこの実施の形態2の要部を示すブロック図である。この実施の形態2では、データ記憶部3を第1のデータ記憶部とし、この第1のデータ記憶部3とは別に第2のデータ記憶部7を設けている。第1のデータ記憶部3としては、データの読み出しと書き込みとの切り替えに際して比較的待ち時間が生じる安価な大容量メモリを使用する。第2のデータ記憶部7としては、データの読み出しと書き込みとの切り替えに際して殆ど待ち時間の生じない高性能の小容量メモリを使用する。また、主演算部2は大容量の第1のデータ記憶部3に対してではなく、小容量の第2のデータ記憶部7に対してアクセス可能に設ける。
【0049】この画像処理装置では次のようにしてFFT処理が行われる。主制御部1は、主演算部2でのFFT処理の前に、第1のデータ記憶部3に格納されている処理すべきデータに対してパイプラインド関数演算部5−1〜5−Nを利用して前処理を行わせる。
【0050】この場合、主制御部1は、関数制御部6に対して、パイプラインド関数演算部5−1〜5−Nの中から前処理に際して実行すべきパイプラインド関数演算部を第1のパイプラインド関数演算群として選択のうえ、その実行順序を指定する。
【0051】関数制御部6は、主制御部1によって指定された実行順序に従って第1のパイプラインド関数演算群のパイプラインド関数演算部を縦続接続させ、第1のデータ記憶部3から処理すべき1番目のデータを読み出して第1のパイプラインド関数演算群に入力する。この第1のパイプラインド関数演算群に入力されたデータは、縦続接続されたパイプラインド関数演算部により順次演算処理され、この第1のパイプラインド関数演算群からの演算処理結果のデータが前処理完了データとして第2のデータ記憶部7に書き込まれる。
【0052】ここで、関数制御部6は、第1のデータ記憶部3から処理すべき1番目のデータを読み出した後、次のサイクルで2番目のデータを読み出して第1のパイプラインド関数演算群に投入する。第1のパイプラインド関数演算群に投入された2番目のデータは、1番目のデータと同様にして、縦続接続されたパイプラインド関数演算部により順次演算処理され、この第1のパイプラインド関数演算群からの演算処理結果のデータが前処理完了データとして第2のデータ記憶部7に書き込まれる。
【0053】第2のデータ記憶部7に2つの前処理完了データが溜まると、主演算部2は、主制御部1の指示に従い、第2のデータ記憶部7に格納されている2つの前処理完了データを読み出し、この前処理完了データに対してFFT処理を施し、このFFT処理を施したデータ（FFT完了データ）を第2のデータ記憶部7に書き込む。
【0054】第2のデータ記憶部7における前処理完了データのFFT処理が終了すると、主制御部1は、関数制御部6に対して、パイプラインド関数演算部5−1〜5−Nの中から後処理に際して実行すべきパイプラインド関数演算部を第2のパイプラインド関数演算群として

選択のうえ、その実行順序を指定する。

【００５５】関数制御部６は、主制御部１によって指定された実行順序に従って第２のパイプラインド関数演算群のパイプラインド関数演算部を縦続接続させ、前処理と同様にして、第２のデータ記憶部７に格納されているＦＦＴ完了データを第２のパイプラインド関数演算群へ次々に投入し、後処理を行わせた後、第１のデータ記憶部３に書き込む。

【００５６】そして、関数制御部６は、第１のデータ記憶部３に全ての後処理完了データを書き込んだ後、主制御部１によって指定される実行順序に従って第１のパイプラインド関数演算群のパイプラインド関数演算部を縦続接続させ、第１のデータ記憶部３に格納されている処理すべきデータの第１のパイプラインド関数演算群への投入を再開する。

【００５７】以下、同様にして、第１のパイプラインド関数演算群を用いての前処理、主演算部２でのＦＦＴ処理、第２のパイプラインド関数演算群を用いての後処理を繰り返すことによって、第１のデータ記憶部３に格納されている全ての処理すべきデータに対して前処理→ＦＦＴ処理→後処理を施す。

【００５８】この実施の形態２では、第１のデータ記憶部３において、データの読み出しと書き込みとを短時間で切り替える必要がなく、第１のデータ記憶部３としてデータの読み出しと書き込みとの切り替えに際して比較的待ち時間が生じる安価な大容量メモリを使用することができる。

【００５９】また、この実施の形態２では、第２のデータ記憶部７は主演算部２でのＦＦＴ処理のためにデータの読み出しと書き込みとが短時間に切り替わる場合でも殆ど待ち時間の生じない高性能のメモリを必要とするが、第２のデータ記憶部７に格納された前処理完了データは主演算部２によってＦＦＴ処理が施された後、ＦＦＴ完了データとして第２のデータ記憶部７に格納されるものの、すぐに関数制御部６によって読み出され第２のパイプラインド関数演算群に投入されるので、第２のデータ記憶部７のメモリ容量は小容量でよい。

【００６０】これにより、第２のデータ記憶部７の追加によるコストアップ分が第１のデータ記憶部３のコストダウン分に吸収され、コストパフォーマンスが向上する。また、この実施の形態２では、第１のデータ記憶部３に他の装置からアクセスがある場合（例えば、画像入力部からの画像データ入力や画像出力部への画像データの出力など）でも、主演算部２での待ち時間が発生せず、処理速度が劣化しない。

【００６１】〔実施の形態３〕実施の形態２では、第２のデータ記憶部７に前処理完了データを書き込んでいる間や第２のデータ記憶部７からＦＦＴ完了データを読み出している間は、主演算部２でのＦＦＴ処理を実行することができず、全演算処理に要する時間が長くなる。そこで、この実施の形態２では、データの転送時間の無駄を削減して、全演算処理に要する時間を短縮する。

【００６２】図３はこの実施の形態３の要部を示すブロック図である。この実施の形態３では、データ記憶部３を第１のデータ記憶部とし、この第１のデータ記憶部３とは別に第２のデータ記憶部７と第３のデータ記憶部８を設けている。第１のデータ記憶部３としては、データの読み出しと書き込みとの短時間の切り替えに際して比較的待ち時間が生じる安価な大容量メモリを使用する。第２のデータ記憶部７および第３のデータ記憶部８としては、データの読み出しと書き込みとの短時間の切り替えに際して殆ど待ち時間の生じない高性能の小容量メモリを使用する。また、主演算部２は大容量の第１のデータ記憶部３に対してではなく、小容量の第２のデータ記憶部７と第３のデータ記憶部８に対してアクセス可能に設ける。

【００６３】この画像処理装置では次のようにしてＦＦＴ処理が行われる。主制御部１は、主演算部２でのＦＦＴ処理の前に、データ記憶部３に格納されている処理すべきデータに対してパイプラインド関数演算部５－１～５－Ｎを利用して前処理を行わせる。

【００６４】この場合、主制御部１は、関数制御部６に対して、パイプラインド関数演算部５－１～５－Ｎの中から前処理に際して実行すべきパイプラインド関数演算部を第１のパイプラインド関数演算群として選択のうえ、その実行順序を指定する。

【００６５】関数制御部６は、主制御部１によって指定された実行順序に従って第１のパイプラインド関数演算群のパイプラインド関数演算部を縦続接続させ、第１のデータ記憶部３から処理すべき１番目のデータを読み出して第１のパイプラインド関数演算群に入力する。この第１のパイプラインド関数演算群に入力されたデータは、縦続接続されたパイプラインド関数演算部ににより順次演算処理され、この第１のパイプラインド関数演算群からの演算処理結果のデータが前処理完了データ（１番目の前処理完了データ）として第２のデータ記憶部７に書き込まれる（図４（ａ）参照）。

【００６６】ここで、関数制御部６は、第１のデータ記憶部３から処理すべき１番目の処理データを読み出した後、次のサイクルで２番目の処理すべきデータを読み出して第１のパイプラインド関数演算群に投入する。第１のパイプラインド関数演算群に投入された２番目のデータは、１番目のデータと同様にして、縦続接続されたパイプラインド関数演算部により順次演算処理され、この第１のパイプラインド関数演算群からの演算処理結果のデータが前処理完了データ（２番目の前処理完了データ）として第３のデータ記憶部８に書き込まれる（図４（ｂ）参照）。

【００６７】一方、主演算部２は、主制御部１からの指示に従い、第２のデータ記憶部７に格納されている１番

目の前処理完了データを読み出し、この１番目の前処理完了データに対してＦＦＴ処理を施す。すなわち、この場合、１番目の前処理完了データに対してＦＦＴ処理が行われている間に、空いている転送経路を利用して、２番目の前処理完了データが先読みされて第３のデータ記憶部８に書き込まれることになる。

【００６８】次に、主演算部２は、主制御部１からの指示に従い、第３のデータ記憶部８に格納されている２番目の前処理完了データを読み出し、この２番目の前処理完了データに対してＦＦＴ処理を施す。この２番目の前処理完了データに対してＦＦＴ処理が行われている間に、関数制御部６は、空いている転送経路を利用して、第２のデータ記憶部７に格納されている１番目のＦＦＴ完了データを第２のパイプラインド関数演算群に投入し、後処理を行わせて、第１のデータ記憶部３に１番目の後処理完了データとして書き込む。そして、この後、関数制御部６は、第１のデータ記憶部３に格納されている３番目の処理すべきデータを第１のパイプラインド関数演算群に投入し、前処理を行わせて、第２のデータ記憶部７に３番目の前処理完了データとして書き込む（図４（ｃ）参照）。

【００６９】次に、主演算部２は、主制御部１からの指示に従い、第２のデータ記憶部７に格納されてい３番目の前処理完了データを読み出し、この３番目の前処理完了データに対してＦＦＴ処理を施す。この３番目の前処理完了データに対してＦＦＴ処理が行われている間に、関数制御部６は、空いている転送経路を利用して、第３のデータ記憶部８に格納されている２番目のＦＦＴ完了データを第２のパイプラインド関数演算群に投入し、後処理を行わせて、第１のデータ記憶部３に２番目の後処理完了データとして書き込む。そして、この後、関数制御部６は、第１のデータ記憶部３に格納されている４番目の処理すべきデータを第１のパイプラインド関数演算群に投入し、前処理を行わせて、第３のデータ記憶部８に４番目の前処理完了データとして書き込む（図４（ｄ）参照）。

【００７０】以下、同様にして、図４（ｃ），図４（ｄ）の処理を交互に行い、第１のパイプラインド関数演算群を用いての前処理、主演算部２でのＦＦＴ処理、第２のパイプラインド関数演算群を用いての後処理を繰り返すことによって、第１のデータ記憶部３に格納されている全ての処理すべきデータに対して前処理→ＦＦＴ処理→後処理を施す。

【００７１】この実施の形態３では、主演算部２と第３のデータ記憶部８との間でＦＦＴ処理を行っている間に、第２のデータ記憶部７からの第２のパイプラインド演算関数群を介する第１のデータ記憶部３への後処理完了データの書き込み、第１のデータ記憶部３からの第１のパイプラインド演算関数群を介する第２のデータ記憶部７への前処理完了データの書き込みを行うことができる。

【００７２】また、主演算部２と第２のデータ記憶部７との間でＦＦＴ処理を行っている間に、第３のデータ記憶部８からの第２のパイプラインド演算関数群を介する第１のデータ記憶部３への後処理完了データの書き込み、第１のデータ記憶部３からの第１のパイプラインド演算関数群を介する第３のデータ記憶部８への前処理完了データの書き込みを行うことができる。

【００７３】これにより、データの転送時間の無駄が削減され、主演算部２でのＦＦＴ処理が休みなく行われるものとなり、全演算処理にかかる時間が短縮されるようになる。

【００７４】また、この実施の形態３では、第１のデータ記憶部３において、データの読み出しと書き込みとの切り替えが頻繁に繰り返されるように思われるが、「前処理のデータ転送時間＋後処理のデータ転送時間」＜主演算時間とすれば、第１のデータ記憶部３でのデータの読み出しと書き込みとを短時間で切り替える必要はなく、第１のデータ記憶部３としてデータの読み出しと書き込みとの短時間の切り替えに際して比較的待ち時間が生じる安価な大容量メモリを使用することができる。

【００７５】また、この実施の形態３では、第２のデータ記憶部７（第３のデータ記憶部８）は主演算部２でのＦＦＴ処理のためにデータの読み出しと書き込みとを短時間で切り替える場合でも殆ど待ち時間の生じない高性能のメモリを必要とするが、第２のデータ記憶部７および第３のデータ記憶部８に格納された前処理完了データは主演算部２によってＦＦＴ処理が施された後、ＦＦＴ完了データとして第２のデータ記憶部７および第３のデータ記憶部８に格納されるものの、すぐに関数制御部６によって読み出され第２のパイプラインド関数演算群に投入されるので、第２のデータ記憶部７および第３のデータ記憶部８のメモリ容量は小容量でよい。

【００７６】なお、この実施の形態３では、説明を簡単とするために、第１のデータ記憶部３から１つずつデータが読み出され、前処理された後、互い違いに第２のデータ記憶部７および第３のデータ記憶部８へ書き込まれるものとしたが、実際にはその投入時間を１サイクルずらした２つのデータがペアとして前処理された後、互い違いに第２のデータ記憶部７および第３のデータ記憶部８へ書き込まれる。主演算部２はこの２つの前処理完了データを読み出してＦＦＴ処理を施す。また、その投入時間を１サイクルずらした２つのＦＦＴ完了データがペアとして第２のデータ記憶部７および第３のデータ記憶部８から互い違いに読み出され、後処理された後、第１のデータ記憶部３に書き込まれる。

【００７７】〔実施の形態２と実施の形態３の処理時間の比較〕実施の形態２（図２）において、第１のデータ記憶部３から第２のデータ記憶部７への転送をＴＲ、第２のデータ記憶部７から第１のデータ記憶部３への転送

をTW、主演算部２でのＦＦＴ処理をＦＦＴとすると、その処理状況は図５（ａ）に示すようになる。

【００７８】実施の形態３（図３）において、第１のデータ記憶部３から第２のデータ記憶部７および第３のデータ記憶部８への転送をＴＲ、第２のデータ記憶部７および第３のデータ記憶部８から第１のデータ記憶部３への転送をＴＷ、主演算部２でのＦＦＴ処理をＦＦＴとすると、その処理状況は図５（ｂ）に示すようになる。

【００７９】図５（ａ）でも図５（ｂ）でもＦＦＴ，ＴＲ，ＴＷの時間幅はそれぞれ同じとする（但し、ＴＲ＋ＴＷ＜＜ＦＦＴ）。すると、ＦＦＴ１～ＦＦＴ１２８（２次元ＦＦＴの半分の時間（横方向か縦方向のみ））の時間は、図５（ａ）の場合には１２８（ＴＲ＋ＦＦＴ＋ＴＷ）、図５（ｂ）の場合にはＴＲ＋１２８ＦＦＴ＋ＴＷとなる。

【００８０】この場合、図５（ａ）と図５（ｂ）との差は１２７ＴＲ＋１２７ＴＷとなり、おおよそ１２７ＴＲ＋１２７ＴＷだけ図５（ｂ）、すなわち実施の形態３の方が高速に処理できることになる。

【００８１】パイプライン関数の処理時間はＴＲ，ＴＷにほとんど含まれてしまう（実際は各パイプライン関数の段数の和だけＴＲ，ＴＷが増加する）。例えば、転送処理に２５６サイクルかかって、パイプライン関数に乗算（３段パイプライン）、加算（２段パイプライン）が転送と同時に縦続接続処理されているとすると、ＴＲ'＝ＴＲ＋３＋２＝２６１サイクルという具合になる。

【００８２】なお、実施の形態１～３では、説明を簡単とするためにパイプラインド関数演算部５－１～５－Ｎは１入力１出力としたが、一部の関数は２入力１出力、３入力１出力などとすることができる（例えば、２画像の差分を算出するような関数）。この場合、第１のパイプラインド関数演算群や第２のパイプラインド関数演算群において、その先頭のパイプラインド関数演算部のみを複数入力１出力とする。複数入力１出力とした場合、他の入力を待つ必要があるので、投入待ち時間Ｘは、入力数をｍとした場合、ＭＡＸ（Ｘｍ）となる。

【００８３】また、実施の形態１～３では、主演算部２においてＦＦＴ処理を行うものとしたが、拡大，縮小，回転，ＤＣＴ，相関計算など各種の主演算に置き換えることが可能である。また、主演算は、パイプライン処理可能な演算であってもよい。しかし、完全にパイプライン処理できるのであれば、それは主演算ではなくパイプラインド関数演算群で実現可能である。ＦＦＴ処理の場合は、全体で考えると１入力１出力で処理することができず、途中で複数経路に分かれるなど複雑な処理となる。主演算部２としては、ＦＦＴ処理のように、１入力１出力で処理することができない複雑な演算が適している。

【００８４】また、実施の形態１～３は画像処理装置への適用例として説明したが、本発明は画像処理に限られるものではなく、各種の演算処理に適用可能である。また、実施の形態２および３では、処理すべきデータに対して前処理と後処理の両方を施す場合を示したが、どちらか一方だけの場合でもよい。

【００８５】

【発明の効果】以上説明したことから明らかなように本発明によれば、第１発明では、データ記憶部から処理すべきデータが読み出され、パイプラインド関数演算群に入力され、このパイプラインド関数演算群からの演算処理結果のデータがデータ記憶部に書き込まれるものとなり、ここで、パイプラインド関数演算群は、そのパイプラインド関数演算群を構成するパイプラインド関数部およびその実行順序が主制御部により指定され、この指定された実行順序に従ってパイプラインド関数演算群のパイプラインド関数演算部が縦続接続されるので、パイプライン関数演算部での処理時間Ｓｎより遥かに小さい投入待ち時間Ｘ（Ｘ＜＜Ｓｎ）でもってパイプラインド関数演算群へ次々にデータを入力することができ、多数の関数を用いる多大なデータ処理を高速で行うことができるようになる。

【００８６】第２発明では、第１のデータ記憶部から処理すべきデータが読み出され、第１のパイプラインド関数演算群に入力され、この第１のパイプラインド関数演算群からの演算処理結果のデータが第２のデータ記憶部に書き込まれ、この第２のデータ記憶部に書き込まれたデータが読み出されて主演算が行われ、その演算処理結果のデータが第２のデータ記憶部に書き込まれ、この第２のデータ記憶部に書き込まれた主演算部での演算処理結果のデータが第２のパイプラインド関数演算群に入力され、この第２のイプラインド関数演算群からの演算処理結果のデータが第１のデータ記憶部に書き込まれるものとなり、ここで、第１および第２のパイプラインド関数演算群は、そのパイプラインド関数演算群を構成するパイプラインド関数部およびその実行順序が主制御部により指定され、この指定された実行順序に従ってパイプラインド関数演算群のパイプラインド関数演算部が縦続接続されるので、パイプライン関数演算部での処理時間Ｓｎより遥かに小さい投入待ち時間Ｘ（Ｘ＜＜Ｓｎ）でもって第１および第２のパイプラインド関数演算群へ次々にデータを入力することができ、多数の関数を用いる多大なデータ処理を高速で行うことができるようになる。

【００８７】また、第２発明では、第２のデータ記憶部では主演算部との間でデータの読み出しと書き込みとが短時間で切り替わる可能性があるが、第１のデータ記憶部ではデータの読み出しと書き込みとが短時間で切り替わらず、第１のデータ記憶部としてデータの読み出しと書き込みとが短時間で切り替わる場合に比較的待ち時間が生じる安価な大容量メモリを使用することができる。また、第２のデータ記憶部としてはデータの読み出しと

書き込みの短時間の切り替わりが生じても待ち時間が殆ど生じない高性能のメモリを必要とするが、第2のデータ記憶部に格納された主演算部で処理すべきデータは主演算部によって主演算が施された後、処理演算結果のデータとして第2のデータ記憶部に格納されるものの、すぐに関数制御部によって読み出され第2のパイプライン関数演算群に投入されるので、第2のデータ記憶部のメモリ容量は小容量でよい。これにより、第2のデータ記憶部の追加によるコストアップ分が第1のデータ記憶部のコストダウン分に吸収され、コストパフォーマンスが向上する。

【0088】第3発明では、第1のデータ記憶部から処理すべきデータが読み出され、第1のパイプライン関数演算群に入力され、この第1のパイプライン関数演算群からの演算処理結果のデータがデータ順に応じて互い違いに第2および第3のデータ記憶部に書き込まれ、また、第1および第2のデータ記憶部に書き込まれたデータがデータ順に応じて互い違いに読み出されて主演算部による演算が行われ、この主演算部での演算処理結果のデータが読み出し元のデータ記憶部に書き込まれ、第1および第2のデータ記憶部に書き込まれた主演算部での演算処理結果のデータが、データ順に応じて互い違いに第2のパイプライン関数演算群に入力され、この第2のパイプライン関数演算群からの演算処理結果のデータが第1のデータ記憶部に書き込まれるものとなり、ここで、第1および第2のパイプライン関数演算群は、そのパイプライン関数演算群を構成するパイプライン関数部およびその実行順序が主制御部により指定され、この指定された実行順序に従って第1および第2のパイプライン関数演算群のパイプライン関数演算部が縦続接続されるので、パイプライン関数演算部での処理時間Snより遥かに小さい投入待ち時間X（X<<Sn）でもって第1および第2のパイプライン関数演算群へ次々にデータを入力することができ、多数の関数を用いる多大なデータ処理を高速で行うことができるようになる。

【0089】また、第3発明では、第2および第3のデータ記憶部では主演算部との間でデータの読み出しと書き込みとが短時間で切り替わる可能性があるが、第1のデータ記憶部ではデータの読み出しと書き込みとが短時間で切り替わらず、第1のデータ記憶部としてデータの読み出しと書き込みとの短時間の切り替えに際して比較的待ち時間が生じる安価な大容量メモリを使用することができる。また、第2のデータ記憶部（第3のデータ記憶部）としてはデータの読み出しと書き込みの短時間の切り替わりが生じても待ち時間が殆ど生じない高性能のメモリを必要とするが、第2のデータ記憶部（第3のデータ記憶部）に格納された主演算部で処理すべきデータは主演算部によって主演算が施された後、処理演算結果のデータとして第2のデータ記憶部（第3のデータ記憶部）に格納されるものの、すぐに関数制御部によって読み出され第2のパイプライン関数演算群に投入されるので、第2のデータ記憶部（第3のデータ記憶部）のメモリ容量は小容量でよい。これにより、第2および第3ののデータ記憶部の追加によるコストアップ分が第1のデータ記憶部のコストダウン分に吸収され、コストパフォーマンスが向上する。

【0090】また、第3発明では、主演算部と第3のデータ記憶部（第2のデータ記憶部）との間で主演算処理を行っている間に、第2のデータ記憶部（第3のデータ記憶部）からの第2のパイプライン演算関数群を介する第1のデータ記憶部への演算処理結果のデータの書き込み、第1のデータ記憶部からの第1のパイプライン演算関数群を介する第2のデータ記憶部（第3のデータ記憶部）への演算処理結果のデータの書き込みを行うことができ、データの転送時間の無駄を削減して、全演算処理に要する時間を短縮することができる。

【図面の簡単な説明】

【図1】 本発明の一実施の形態（実施の形態1）の要部を示すブロック図である。

【図2】 本発明の他の実施の形態（実施の形態2）の要部を示すブロック図である。

【図3】 本発明の他の実施の形態（実施の形態3）の要部を示すブロック図である。

【図4】 実施の形態3における処理動作を説明する図である。

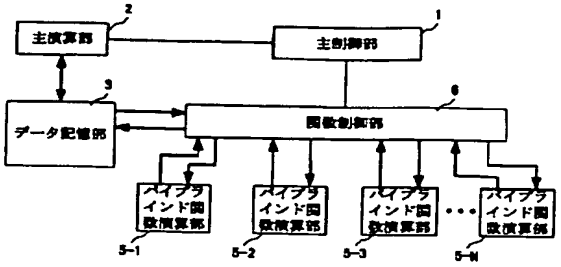
【図5】 実施の形態2と実施の形態3の処理時間の比較を説明する図である。

【図6】 従来の画像処理装置の要部を示すブロック図である。

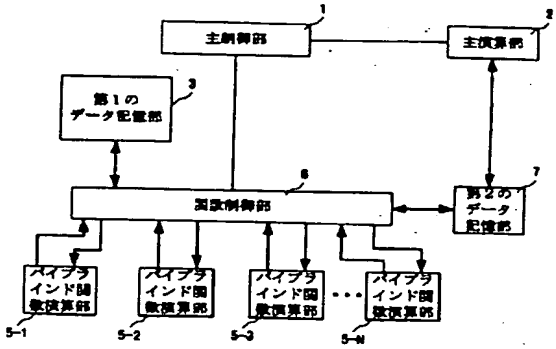
【符号の説明】

1…主制御部、2…主演算部、3…データ記憶部（第1のデータ記憶部）、5－1～5－N…パイプライン関数演算部、6…関数制御部、7…第2のデータ記憶部、8…第3のデータ記憶部。

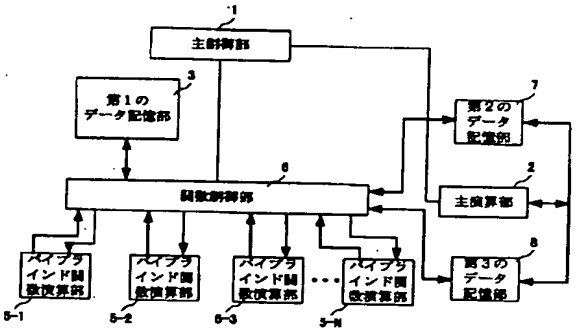
【図1】



【図2】



【図3】



【図5】

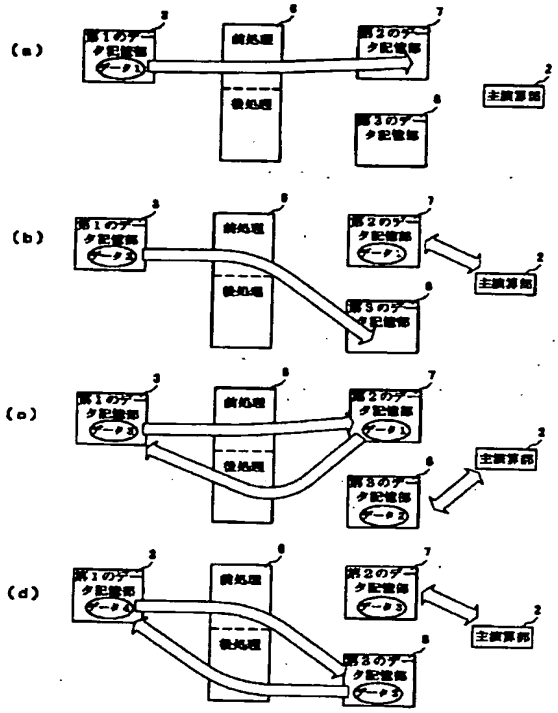
(a)

主演算部		FFT1				FFT2				FFT3				FFT128					
第2のデータ記憶部	TR1	FFT1	TW1	TR2		FFT2	TW2	TR3		FFT3	TW3	…		FFT128	TW128				

(b)

主演算部		FFT1				FFT2				FFT3				FFT128					
第2のデータ記憶部	TR1	FFT1	TW1	TR3		FFT3	TW2	TR4		FFT3	…			TW127					
第3のデータ記憶部	TR2			FFT2					…					FFT128	TW128				

【図4】



【図6】